# When Does Syntax Mediate Neural Language Model Performance? Evidence from Dropout Probes

**Mycal Tucker**
mycal@mit.edu

**Tiwalayo Eisape**
eisape@mit.edu

**Peng Qian**
pqian@mit.edu

**Roger Levy**
rplevy@mit.edu

**Julie Shah**
julie_a_shah@csail.mit.edu

## Abstract

Recent causal probing literature reveals when language models and syntactic probes use similar representations. Such techniques may yield "false negative" causality results: models may use representations of syntax, but probes may have learned to use redundant encodings of the same syntactic information. We demonstrate that models do encode syntactic information redundantly and introduce a new probe design that guides probes to consider all syntactic information present in embeddings. Using these probes, we find evidence for the use of syntax in models where prior methods did not, allowing us to boost model performance by injecting syntactic information into representations.

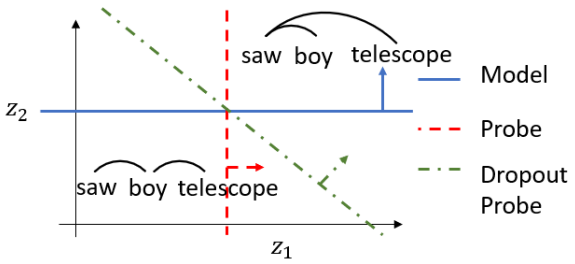



Figure 1: In a 2D embedding space, a model might redundantly encode syntactic representations of a sentence like "the girl saw the boy with the telescope." Redundant encodings could cause misalignment between the model's decision boundary (blue) and a probe's (red). We introduce dropout probes (green) to use all informative dimensions.

## 1 Introduction

Recent large neural models like BERT and GPT-3 exhibit impressive performance on a large variety of linguistic tasks, from sentiment analysis to question-answering (Devlin et al., 2019; Brown et al., 2020). Given the models' impressive performance, but also their complexity, researchers have developed tools to understand what patterns models have learned. In probing literature, researchers develop "probes:" models designed to extract information from the representations of trained models (Linzen et al., 2016; Conneau et al., 2018; Hall Maudslay et al., 2020). For example, Hewitt and Manning (2019) demonstrated that one can train accurate linear classifiers to predict syntactic structure from BERT or ELMO embeddings. These probes reveal what information is present in model embeddings but not how or if models use that information (Belinkov, 2021).

To address this gap, new research in causal analysis seeks to understand how aspects of models' representations affect their behavior (Elazar et al., 2020; Ravfogel et al., 2020; Giulianelli et al., 2018; Tucker et al., 2021; Feder et al., 2021). Typically, these techniques create counterfactual representations that differ from the original according to some property (e.g., syntactic interpretation of the sentence). Researchers then compare outputs when using original and counterfactual embeddings to assess whether a property encoded in the representation is causally related to model behavior.

Unfortunately, negative results — wherein researchers report that models do not appear to use a property causally — are difficult to interpret. Such failures can be attributed to a model truly not using the property (true negatives), or to a failure of the technique (false negatives). For example, as depicted in Figure 1, if a language model encodes syntactic information redundantly (here illustrated in two-dimensions), the model and probe may differentiate among parses along orthogonal dimensions. When creating counterfactual representations with such probes, researchers could incorrectly conclude that the model does not use syntactic information.

In this work, we present new evidence for the causal use of syntactic representations on task performance in BERT, using newly-designed probes that take into account the potential redundancy in a model's internal representation. First, we find evidence for representational redundancy in BERT-based models. Based on these findings, we propose

a new probe design that encourages the probe to use all relevant representations of syntax in model embeddings. These probes are then used to assess if language models use representations of syntax causally, and, unlike prior art, we find that some fine-tuned models do exhibit signatures of causal use of syntactic information. Lastly, having found that these models causally use representations of syntax, we used our probes to boost a question-answering model's performance by "injecting" syntactic information at test time.[1]

## 2 Related Work

### 2.1 Language Model Probing

Probing literature seeks to expose learned patterns of a neural language model by training small neural networks to map from model representations to human-interpretable properties (Alain and Bengio, 2017; Conneau et al., 2018; Coenen et al., 2019). For example, Hewitt and Manning (2019) propose single-layered neural nets that map from embeddings to syntactic representations of sentences. Such probing methods are correlative rather than causal because they depict what information is present in representations instead of how that information is used (Hall Maudslay et al., 2020; Pearl and Mackenzie, 2018). Understanding when language models use structural information causally is an important question given the central role structure appears to play in human understanding of natural language (Chomsky, 1965). In this work, we perform causal analysis by combining causal methods with a new probe design.

### 2.2 Causal Analysis of Language Models

Recently, researchers have begun applying causal analysis to language models to understand if and how they use human-interpretable properties in their decision making. While direct text manipulations are sometimes possible (e.g., modifying "The man works as a..." to "The woman works as a ..."), several methods rely on constructing counterfactual representations to measure model behavior (Kaushik et al., 2020; Ravfogel et al., 2020). Prior art has often found that standard models learn undesirable causal relationships by encoding unwanted biases or by not learning to rely upon syntactic principles (Feder et al., 2021; Elazar et al., 2020).

[1]Code at https://github.com/mycal-tucker/mlm_dropout_probes

Our work is most closely related to Tucker et al. (2021), so we explain their technical approach here. Tucker et al. (2021) train non-linear structural probes (based on those designed by Hewitt and Manning (2019)) to predict aspects of a sentence's syntactic structure from model embeddings. That is, a probe $p$ maps from an embedding, $z$, to a representation of syntax, $s$. Trained probes are used to create counterfactual embeddings, $z'$, by updating $z'$ from $z$ via gradient descent to minimize a loss function, $L$, evaluated on the probe's output and a desired output based on an alternative syntactic interpretation, $s'$: $\nabla_{z'} L(p(z'), s')$. Intuitively, these $z'$ are meant to represent "what $z$ would have been if the structure of the sentence were $s'$." Using suites of syntactically ambiguous sentences, the authors measured how a model's outputs differed when using $z'$ generated from different syntactic interpretations.

While Tucker et al. (2021) find that a pretrained BERT model does use representations of syntax causally (i.e., model outputs change when using different syntactic interpretations), the authors find that a BERT model fine-tuned on a question-answering task does *not* show similar behavior. Identifying causal mechanisms in models is important not only for fairness and robustness measures, but also for improving model performance. In specific cases of subject-verb agreement, Giulianelli et al. (2018) found that changing representations of a subject's plurality affected the plurality of verbs predicted by an LSTM.

In this work, we use the gradient-descent method proposed by Tucker et al. (2021), but we use a new probe design. We identify several cases in which their method fails to uncover a causal relationship, whereas ours does. Furthermore, compared to Giulianelli et al. (2018), we use more general representations of syntax instead of only plurality.

## 3 Technical Approach

Here, we identify a limitation of prior causal probing art in which redundant information in embeddings could lead to probes and models using different representations of the same information, which in turn could lead to uninformative causal analysis results. We propose a new probe architecture that addresses this limitation by encouraging probes to use all sources of information in embeddings.

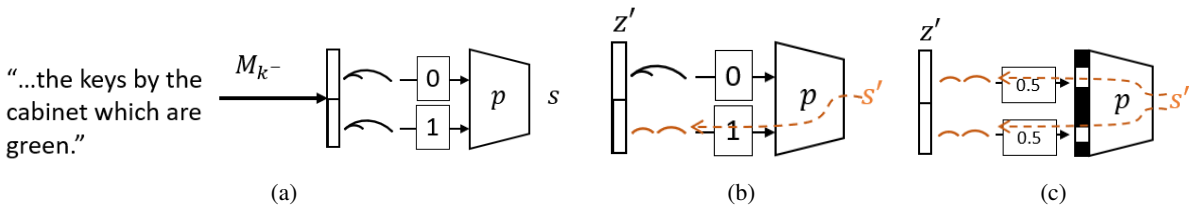



Figure 2: If a model encodes the dependency structure of a sentence twice its embedding, a probe, $p$, may learn to ignore one copy of the information (indicated by learned weight 0) and only use the other (via learned weight 1) to predict $s$ (a). In such cases, the gradient of $s$ with respect to the embedding (dashed orange) only flows from one of the copies, so only that copy will be updated in counterfactual embeddings (b). However, by introducing a dropout layer that masks random inputs to the probe, dropout probes learn to use all informative parts of embeddings, which distributes the gradient across the whole embedding (c).

### 3.1 Limitations from Redundancy

We show by example how prior art in causal probing may fail to reveal causal uses of syntactic information in language models. Here, we use a simplified example; in later experiments we demonstrate that trained models exhibit similar phenomena.

In neural network probing literature, a probe, $p_\theta$, is a neural network parametrized by weights, $\theta$, that maps from representations, $z$, to a predicted property, $\hat{s}$: $\hat{s} = p_\theta(z)$. Hewitt and Manning (2019) define two types of structural probes that map from $z$ to representations of a sentence's syntax. The "depth" probe predicts words' depths in a parse tree; the "distance" probe predicts the distance between pairs of words in a parse tree. In this paper, we assume $s$ refers to syntactic information, but probing techniques are general. Given a corpus comprising $(z, s)$ pairs, probes are trained using supervised learning to minimize some loss.

Suppose that there exists a trained model, $M$; $M_{k-}$ (the first $k$ layers of $M$) encodes an input, $x$, into an embedding $z$. The layers of $M$ after $k$, dubbed $M_{k+}$, produce a prediction, $\hat{y}$, from $z$. For the purposes of this example, we state that $M$ uses syntactic information, and specifically that $z$ is informative of the syntactic structure of $x$.

Let us assume that the dependency structure of $x$ may be represented by within a vector, $z_{dep}$, and that $M_{k-}$ produces embeddings, $z$, which are two identical copies of $z_{dep}$. Using pythonic notation, $z = [z_{dep}] + [z_{dep}]$. Thus, $z$ contains syntactic information and, when we state that $M$ "uses" syntactic information, we formally mean that $\nabla_{z_{dep}} M_{k+}(z) \neq 0$.

Building upon this example, let us label the two copies of $z_{dep}$ as $z_{dep^1}$ and $z_{dep^2}$, although the two vectors remain identical. If we train a probe to predict syntactic forms from $z$, it may arbitrarily learn to use any aspects of $z$ that are informative of its prediction, $s$. Let us say that the probe learns to use only $z_{dep^2}$, again defined as $\nabla_{z_{dep^2}} p(z) \neq 0$. However, $M_{k+}$ may only use $z_{dep^1}$: the copy that the probe does not use.

We claim that this example, while simplified, demonstrates a potential scenario in which causal probing techniques could return a false negative. Specifically, if one generates counterfactual embeddings, $z'$, by changing $z$ according to the activations that change the probe's outputs, only $z_{dep^2}$ will change. Because $M_{k+}$ uses only $z_{dep^1}$ for predictions, the model's output will not change. This example is depicted in Figure 2. Ultimately, without considering the redundancy in a model's internal representation, prior methods will fail to uncover the fact that $M$ actually does use representations of syntax causally.

### 3.2 Dropout Probes

In this section, we propose a neural probe architecture to address the limitations of prior art by encouraging probes to use all syntactic information present in $z$. The desired behavior is depicted in Figure 2c: if the probe uses all activations that are informative of syntax, that will necessarily be a superset of the activations that the model uses for downstream processing (if the model uses syntax). Therefore, when generating counterfactual embeddings using such probes, every activation encoding syntactic information would be updated, which in turn would change the model's output.

Our approach was inspired by an idea of creating a mixture of probes, each trained to use a different masked subset of activations in $z$. The full set of such probes would have to learn to use all activations in $z$ that are informative of $s$. One may approximate creating such a set by introducing a

dropout layer as the first layer to a single probe. At training time, the dropout layer masks a random subset of the input; the mask itself changes with every training batch. We dub such probes "dropout probes." This probe design, and our resulting findings when using them, are the main contributions of our work. We note that adding a dropout layer to probes introduces a new hyperparameter but, in experiments, we found consistent results over a wide range of positive dropout values.

## 4 Experiments

Here, we report the results from three experiments establishing the benefits of dropout probes. First, we found evidence supporting our hypothesis of redundantly-encoded syntactic information by calculating the mutual information between various activations in trained networks. Second, we compared dropout probes to standard probes in a set of syntactically-ambiguous test domains. We found that our method revealed evidence supporting the causal use of syntax in models where other methods did not (Tucker et al., 2021). Lastly, given our findings that models used syntax causally, we demonstrated how one could "inject" syntactic information into models to improve performance in syntactically-challenging tasks.

Experiments were conducted on four models, all based on huggingface's `bert-base-uncased` (Wolf et al., 2019). The Mask model was the original model, trained on a masked language modeling task and next-sentence prediction (Devlin et al., 2019). The QA model was fine-tuned on the Stanford Question Answering Dataset 2.0 (Rajpurkar et al., 2016).[2] Lastly, we trained two models, dubbed NLI and NLI-HANS, that were finetuned on the Multi-Genre Natural Language Inference dataset or that dataset augmented with the Heuristic Analysis for NLI Systems (HANS) dataset, respectively (Williams et al., 2018; McCoy et al., 2019).

The Mask model was used to compare our method to Tucker et al. (2021), who found that such models used syntactic information causally. The QA model was used to study a finetuned model; prior art did not find evidence of causal use. Lastly, the NLI models were used because Natural Language Inference is recognized as a difficult linguistic task that models appear to "cheat" on by leveraging spurious correlations in datasets (McCoy et al., 2019; Naik et al., 2018; Sanchez et al., 2018).

[2]The QA model was downloaded from huggingface model repository under "twmkn9/bert-base-uncased-squad2"

| | $I(Z_1, D)$ | $I(Z_2, D)$ | $I(Z, D)$ |
|---|---|---|---|
| Mask | 2.2 | 2.6 | 2.7 |
| QA | 2.7 | 2.8 | 2.8 |
| NLI | 2.3 | 2.7 | 2.8 |

Table 1: The mean in nats of $I(Z, D)$ is less than $I(Z_1, D) + I(Z_2, D)$, indicating that information about $D$ is redundantly encoded in embeddings. Standard deviation under 0.2 for all values over 5 trials.

### 4.1 Measuring Redundancy in Embeddings

First, we found that language models redundantly encoded syntactic information in their embeddings, which motivated using dropout probes.

We used a technique from prior art, Mutual Information Neural Estimator (MINE), which is a neural-network based approach for estimating the mutual information between two random variables (Belghazi et al., 2018). It does so by computing a lower bound of mutual information and training a neural network to maximize that value. This provides a conservative but tight estimate of mutual information. We refer readers to Appendix A for further details of our implementation.

We defined four random variables of interest. The first, $D$, was the depth of each word in a sentence's parse tree; in other words, the labels used to train depth probes in prior literature (Hewitt and Manning, 2019). The second random variable, $Z$, was the 768-dimensional embeddings generated by a language model for each token in an input sentence. Lastly, the third and fourth random variables ($Z_1$ and $Z_2$) corresponded to the first and second halves of $Z$ for each token. That is, these variables comprised the starting and ending 384 units for each token's embedding. By measuring the mutual information between different pairs of these variables, one may formalize our redundancy hypothesis into the following test: $I(Z, D) < I(Z_1, D) + I(Z_2, D)$. Intuitively, if the test holds, there is shared syntactic information between $Z_1$ and $Z_2$.

We trained a MINE neural network on the first 5000 examples from the Penn TreeBank to estimate mutual information between random variables (Marcus et al., 1993). Embeddings were taken from the fourth layer of the MASK, QA, and NLI models, although they may be generated elsewhere. Our results are presented in Table 1. For all models, $I(D, Z) < I(D, Z_1) + I(D, Z_2)$; i.e., one gains

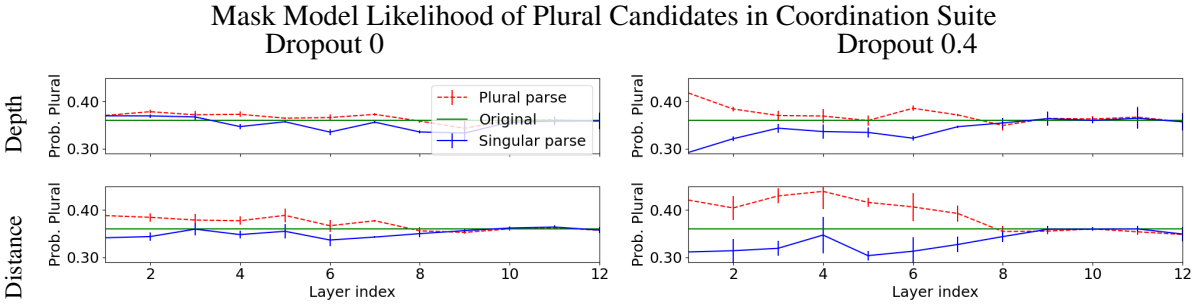



Figure 3: Mean and standard deviation probabilities over 5 trials for plural candidates using the original embeddings (green) or counterfactual embeddings favoring plural (dashed red) or singular (solid blue) parses. Counterfactual embeddings generated by both depth- and distance-based probes caused the greatest shift in model predictions.

little to no information for predicting $D$ from the full $Z$ instead of from just $Z_1$ or just $Z_2$. This is evidence of redundant syntactic information in $Z$.

In these experiments using MINE, we demonstrated how $Z_1$ and $Z_2$ could be defined as the subsets of redundant activations depicted in Figure 2. One could define other $Z_1$ and $Z_2$ to better characterize redundancy; here, we merely claim that at least some redundancy is present in the model embeddings.

### 4.2 Ambiguity Suite Experiments

The prior section established that language models encode syntactic information redundantly; here, we showed that dropout probes overcame the challenges introduced by this redundancy by better aligning with models' true causal usage of syntax. We compared dropout probes to the probes used in prior art via counterfactual experiments inspired by those used by Tucker et al. (2021).

We trained both distance- and depth-based probes, the two types of syntactic probes proposed by Hewitt and Manning (2019). We trained a new probe for each layer of each model, conducting 5 trials with random seeds 0 through 4. All probes were implemented as 3-layer, non-linear neural nets that mapped from model embeddings (of dimension 768) through 2 ReLU layers of dimension 1024, to a final layer to predict a word's depth or distance in the parse tree from other words. Probes were trained for up to 100 epochs, with early stopping based on validation set loss, using the Penn TreeBank dataset (Marcus et al., 1993). We found that this produced more accurate probes than prior art, which capped training at 30 epochs, and that these resulting probes did better than prior reported results, even without using dropout. Each probe was prefixed by a dropout layer with a parameter, $\alpha$, that specified the proportion of inputs that were masked before being fed to the probe. By setting $\alpha = 0$, we recreated prior art of standard probes. We additionally investigated positive values of $\alpha$ to measure the benefit of dropout. Counterfactual embeddings were created via gradient descent through trained probes (with dropout disabled), as in prior art (Tucker et al., 2021). That is, new embeddings, $z'$, were generated to decrease the loss between $p(z')$ and a desired parse. We called this loss the counterfactual loss.

In these experiments, we reported two types of results. First, we visualized the effect of interventions, by layer, for a particular dropout rate and counterfactual loss. This revealed that, typically, earlier layers in models were more susceptible to interventions. Second, we devised an aggregate metric for the average difference, across all layers, in model outputs for counterfactuals generated with different parses. This showed how lower counterfactual losses (i.e., more interventions) and higher dropout typically revealed larger effects.

Additionally, we note that the probes were trained to parse single sentences, but two of the models (QA and NLI) accepted two sentences as inputs. For both models, counterfactual embeddings were creating by only updating the syntactically-ambiguous sentence and then concatenating it to the unaltered other embeddings.

#### 4.2.1 Masked Language Model

In testing the Mask model, we largely reproduced patterns in prior results that such models use representations of syntax causally, although we found new results with dropout depth probes. We tested the model with ambiguity test suites inspired by the Coordination and NP/Z suites from Tucker et al. (2021). For example, in the Coordination suite,

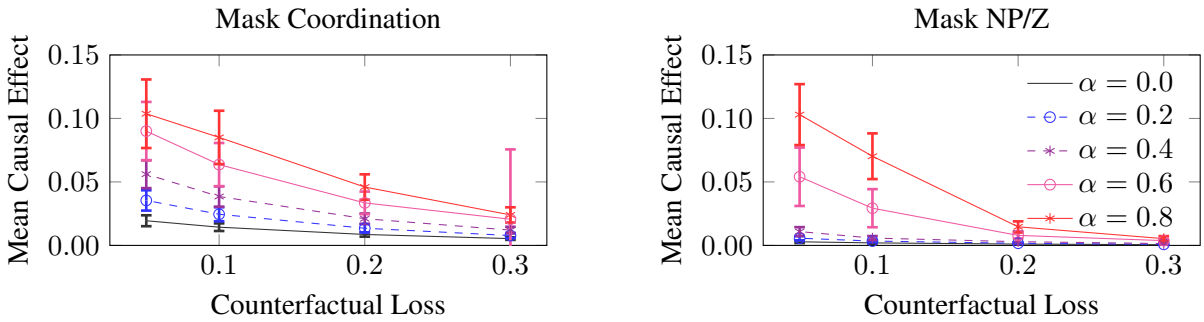



Figure 4: For the Coordination (left) and NP/Z (right) suites, interventions to a lower counterfactual loss ($x$ axis) and with higher-dropout probes (different curves) revealed the greatest causal effects. Means and standard errors.

one sentence reads, "The man saw the girl and the dog [MASK] tall." One may plausibly insert either a plural or singular noun in the masked location, depending upon the syntactic interpretation of the sentence. We generated sentences using a template-based method; details of the prompts (and all prompts in this work) are included in Appendix B.

The results of passing $z'$ generated from different parses in the Coordination suite through the rest of the Mask model are plotted in Figure 3. The three plotted lines correspond to the model output using the normal embeddings (green), using $z'$ generated according to a parse favoring plural verbs (red dashed), or using $z'$ generated using parses implying singular verbs (blue solid). The $y$ axis corresponds to the probability the model assigned to words implying a plural interpretation ("were," "are," and "as") fitting in the masked location, normalized by the sum of probabilities assigned to those plural words or singular words ("was" and "is"). If the Mask model uses syntactic representations correctly, counterfactuals from plural parses should increase the probability of plural words.

We indeed found that effect, although it is clearest when using dropout probes. The causal effects using standard probes are plotted in the left column; we reproduced the findings from prior art that distance-based probes create the desired effect, but depth-based probes had little to no effect. Conversely, when using dropout probes with $\alpha = 0.4$ (right column), we found much larger effects.

Averaging across all layers, we also measured the mean difference in output when using counterfactual embeddings generated according to different parses. Intuitively, this generated a single number that captured the average difference between the red and blue lines in the plots in Figure 3.

For a range of dropout values and counterfactual losses, we plotted the mean causal effect for the Coordination and NP/Z suites in Figure 4, using distance probes. For a given counterfactual loss, using higher dropout probes produced larger effects. In addition, lower counterfactual losses (corresponding to more gradient steps) induced greater effects. These trends also held true for depth-based probes (Appendix D). Overall, using the Mask model, we recreated prior art and found new evidence that models also use a depth-based representation of syntax.

#### 4.2.2 QA Model

We also found that the QA model used representations of syntax causally, contrary to prior findings, through a series of similar causal analysis experiments using syntactically-ambiguous inputs. The QA model is a BERT-based model fine-tuned on a question-answering task to map from context and a question to a continuous span of the context that answered the question (Rajpurkar et al., 2016).

We performed experiments using depth- and distance-based probes, using dropout values at increments of 0.1 from 0 to 0.9. We used three test suites for analyzing the causal use of syntax in the QA model: "Coordination", "Relative Clause" (RC), and a "Noun Phrase/Verb Phrase" (NP/VP) suite. The Coordination suite consisted of 256 prompts with coordination ambiguity like, "I saw the men and the women were tall. Who was tall?" The RC suite consisted of 193 prompts with attachment ambiguity of a relative clause like, "I saw the women and the men who were tall. Who was tall?" The NP/VP suite consisted of 256 prompts like, "The girl saw the boy with the telescope. Who had the telescope?" Prompts were designed such that answers were dictated by syntactic interpretations.

Findings for the Coordination suite are plotted

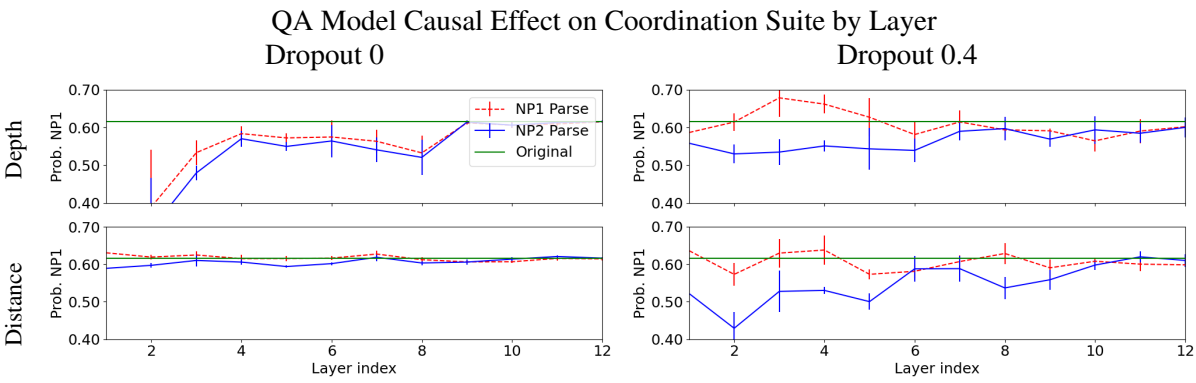



Figure 5: Causal effects for the QA model using depth- (top row) and distance- (bottom row) based probes with dropout of 0 (left column) or 0.4 (right column) on the Coordination corpus to counterfactual loss 0.05. Dropout probes produce more stable and larger effect sizes. Means and standard deviations over 5 trials plotted.

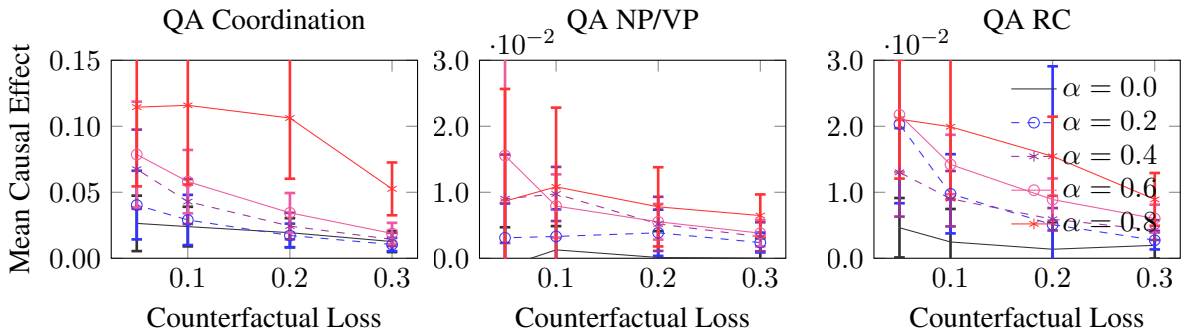



Figure 6: Mean causal effects when using depth-based probes for the QA test suites. Smaller counterfactual losses and higher dropout rates typically induced greater effects, although the scale of the effects varied by suite (note different axis scales). Means and standard deviations over 5 trials.

in Figure 5. On the $y$ axis, we plotted the model's prediction of words in the first noun phrase (NP1) starting the answer. Correct causal use of representations of syntax would move the red line (corresponding to parses indicating NP1) above the original outputs, in green, and the blue line (for the other parse) below.

Unlike prior art, we found evidence that QA models use representations of syntax causally. In the left column of Figure 5, we found similar results to prior art: using standard depth-based probes produced noisy results, and distance-based probes had a small effect. (In fact, this effect size shrank if we only trained the distance probe for 30 epochs, as in prior art, instead of the 100 epochs we used, indicating the importance of well-trained probes.) In contrast to the standard probes, the dropout probes, plotted in the right column, revealed much larger effects of syntactic interventions.

More systematic analysis for all dropout rates, using distance and depth-based probes for all 3 test suites confirmed these trends. We plotted the aggregate metrics for all suites using depth probes in Figure 6. The causal effects were smaller in the RC and NP/VP suites than in the Coordination suite, indicating that the model may have learned a weaker causal link for these syntactic relations. Nevertheless, all suites demonstrate the importance of using dropout in probes: without dropout (solid black curve), the causal effects were smaller than for any positive dropout rate.

We note briefly that the causal effects uncovered by dropout probes may not be solely attributed to dropout probes performing better at their parsing task. In fact, adding dropout worsened probe performance according to typical probe performance metrics (Appendix E).

#### 4.2.3 NLI Model

Lastly, we performed similar causal analysis on the NLI and NLI-HANS models and, in contrast to the Mask and QA models, we found no evidence for the causal use of syntax using any of our probes for either model. The NLI model was finetuned on just the MNLI corpus, and the NLI-HANS model was finetuned with both the MNLI and HANS corpora,

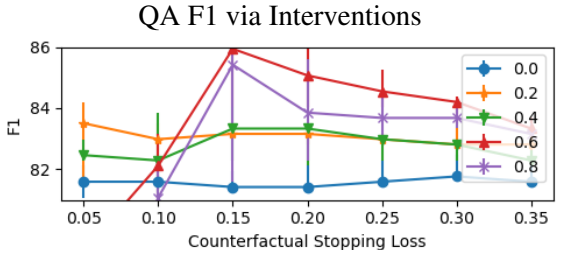



Figure 7: Using dropout probes over a range of dropout values (different curves) and counterfactual stopping losses improved model performance, and dropout typically improved performance. Medians and quartiles plotted over 5 trials.

based on code from Gao et al. (2021). The NLI model had a test set accuracy of 86%, and the NLI-HANS model had test set accuracy of 93%.

We used a test suite based on the Coordination suite already introduced in this work: an example prompt was "The person saw the keys in the cabinets which are green. The keys are green." The models had to classify these inputs among three classes of entailment, contradiction, or neutrality.

Ultimately, we failed to find any evidence that either the NLI or the NLI-HANS model used syntactic information causally. The models always predicted entailment for all prompts, whether using original embeddings or counterfactuals generated for different parses. We used distance probes with dropout values from 0 to 0.9 and created counterfactuals for losses from 0.05 to 0.3 and never observed a shift in predicted probability mass of more than 1% when using counterfactuals. Unfortunately, this suggests that simply augmenting the MNLI dataset with HANS may not be enough to produce a model that uses syntactic information causally.

### 4.3 Boosting Performance with Probes

Earlier, we demonstrated that the QA model causally used representations of syntax for predictions; here, we showed that we could improve QA model performance at test time by "injecting" syntactic information into embeddings. Because prior art had not found that QA models used syntax causally, such interventions were not previously pursued, as far as we are aware.

We designed a new, syntactically challenging "Intervene" test suite of 288 prompts for the QA model. Example prompts are "The person saw the keys by the cabinet which was green. What was green?" and "The person saw the keys by the cabinet which were green. What was green?" Answering correctly ("the cabinet" first and "the keys" second) depends upon using noun-verb agreement. We used template-generated parse trees for each sentence and distance probes to create counterfactual embeddings for each sentence at layer 4 of the QA model. Layer 4 was chosen based on performance on a validation dataset (Appendix C).

We passed the original and counterfactual embeddings through the QA model and measured performance on a test suite. F1 performance is plotted in Figure 7; exact match metrics had similar trends. Typically, higher-dropout probes improved performance more, although the highest-dropout probes deteriorated for the lowest counterfactual losses. We hypothesize that this deterioration corresponded to generating out-of-distribution embeddings, but this topic warrants further study.

Lastly, we performed a similar experiment using the NLI and NLI-HANS models using 486 prompts drawn from the HANS dataset like "The doctor near the actor danced. The actor danced" (McCoy et al., 2019). The NLI model achieved 50% accuracy (always predicting entailment) and the NLI-HANS model achieved 99% accuracy. Neither model's accuracy changed significantly when using counterfactuals with the correct parse for the first sentence, yet again indicating that these models may not use representations of syntax causally.

## 5 Contributions and Conclusion

In this work, we designed and evaluated "dropout probes," a new neural probing architecture for generating useful causal analysis of trained language models. Our technical contribution — adding a dropout layer before probes — was inspired by a theory of redundant syntactic encodings in models. Our results fit within three categories: we showed that 1) models encoded syntactic information redundantly, 2) dropout probes, unlike standard probes, revealed that QA models used syntactic representations causally, and 3) by injecting syntactic information at test time in syntactically-challenging domains, we could increase model performance without retraining.

Despite our step towards better understanding of pretrained models, future work remains. Natural extensions include studying pretrained models beyond those considered in this work, further research into redundancy in embeddings, more investigation into inserting symbolic knowledge into neural representations, and new methods for training models

to respond appropriately to interventions.

## 6 Ethical and Broader Impacts

While the majority of this paper details the technical contributions of our work, here, we briefly consider some of the possible consequences of our findings based on transparency and causal modeling.

Fundamentally, we believe that causal analysis of models is a powerful tool towards more ethical AI. Our dropout probes enable better inspection of models, providing possible mechanisms for regulators, ethicists, and even the general public to better understand AI systems with which they interact. By injecting information into models at test time, as demonstrated in Section 4.3, we provide another mechanism for people to control model behavior. Thus, our tool may reinforce values of transparency and value-alignment in AI, contingent upon access to the model for probing.

While we hope that our probing mechanism will be used for good, misuse of the tool is certainly possible. In particular, the very causal rules that our tool uncovers may be used to reinforce biases. For example, people may attempt to argue that a gender bias exhibited by a model are evidence of the "correctness" of that bias. We urge readers to remember that models likely reflect biases present in human-generated data and certainly not "true" stereotypes.

We also note that the transparency benefits of our technique are not universally accessible. Training a single probe on a single layer took approximately 2 minutes on an NVIDIA GeForce 3080; generating counterfactuals took approximately 1 second per counterfactul on similar hardware. Although these operations individually are relatively lightweight (and certainly less computationally intensive than finetuning a whole model), systematic evaluation of models for many layers, multiple probes, and many counterfactuals is more challenging. Furthermore, all analysis assumes access to the internals of the pretrained model itself.

Lastly, while our work is limited to diagnosis of existing models, we hope that it will enable important future research in causally-motivated models. We hope to ultimately develop models that blend causal rules based on human guidance with emergent learned patterns from data. Our work can complement such research by certifying that models have indeed learned the desired rules.

## 7 Acknowledgements

TE acknowledges support from the GEM consortium and the National Science Foundation Graduate Research Fellowship under Grant No. 1745302.

## References

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. *ArXiv*, abs/1610.01644.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 531–540. PMLR.

Yonatan Belinkov. 2021. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, pages 1–13.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT press.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8594–8603. Curran Associates, Inc.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *arXiv e-prints*, page arXiv:2006.00995.

Amir Feder, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386.

Yang Gao, Nicolò Colombo, and Wei Wang. 2021. Adapting by pruning: A case study on BERT. *CoRR*, abs/2105.03343.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*, 1st edition. Basic Books, Inc., USA.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7237–7256, Online. Association for Computational Linguistics.

Ivan Sanchez, Jeff Mitchell, and Sebastian Riedel. 2018. Behavior analysis of NLI models: Uncovering the influence of three factors on robustness. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1975–1985, New Orleans, Louisiana. Association for Computational Linguistics.

Mycal Tucker, Peng Qian, and Roger Levy. 2021. What if this modified that? syntactic interventions with counterfactual embeddings. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 862–875, Online. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

## A MINE Details

The Mutual Information Neural Estimator technique works by training a neural net to compute and maximize a lower bound on mutual information between two random variables (Belghazi et al., 2018). We describe the intuition of the technique, as well as our implementation, in this section; we refer readers to the full paper for theoretical analysis of MINE.

The mutual information between two variables is defined via the KL Divergence between the joint distribution of the variables and

the product of their marginals: $I(X,Y) = D_{KL}(P(XY)||P(X)P(Y))$. For notational simplicity, we describe the joint distribution as $P$ and the product of the marginals as $Q$. Let us further state that $P$ and $Q$ define outputs that are jointly in $R^D$.

A lower-bound for the KL divergence is as follows, setting $F$ as any class of functions that maps from $R^D$ to $R$:

$$D_{KL}(P||Q) \geq \sup_{T \in F} \mathbb{E}_P[T] - (\mathbb{E}_Q[e^T]) \quad (1)$$

In other words, one can lower bound the mutual information by finding a function, $T$, that maximizes the difference between the two terms in Equation 1. Belghazi et al. (2018) do so with functions parametrized as a neural net that maps from the concatenation of two inputs (one for each random variable) to a single-valued output. Training the neural net is conducted to maximize the value described by Equation 1.

In our experiments, we create neural networks with separate, linear layers of size 64 for each input. The embeddings from those two layers are concatenated, passed through two 1024-dimensional layers with ReLU activations, and then passed through a linear layer with a single output. We thus mapedp from the two inputs to a single, real-valued output.

Training was performed using batch size 32 over 50 epochs, at which point the mutual information estimates appeared to have converged.

## B Test Suite Creation

Here, we specify the details of the test suites used to evaluate models for reproducibility.

The Mask model Coordination test suite comprised sentences like "The man saw the girl and the dog [MASK] tall." More generally, sentences followed the following template: "The NN1 V the NN2 and the NN3 [MASK] ADJ." We created all sentences by iterating through the combinations of the words described in Table 2. This generated 243 sentences, and each sentence was associated with 2 parses: one described as a conjunction of sentences (e.g., "(The man saw the girl) and (the dog [MASK] tall.)") and one as a single sentence with a conjunction of noun phrases (e.g., "The man saw (the girl and the dog) [MASK] tall.").

The mask model NP/Z test suite comprised sentences like, "When the dog scratched the vet [MASK] ran." More generally, sentences followed the following template: "When the NN1 V1 the NN2 [MASK] V2." Each sentence was associated with two parses, favoring either adverbs (e.g., "When the dog scratched the vet quickly ran" or nouns, "When the dog scratch the vet she ran"). We used the word tuples described in Table 3, inspired by prior art, to generate 150 sentences.

| Category | Words |
|---|---|
| NN1 | man, woman, child |
| NN2 | boy, building, cat |
| NN3 | dog, girl, truck |
| V | saw, feared, heard |
| ADJ | tall, falling, orange |

Table 2: Words used for sentence generation in the Mask Coordination test suite.

The QA model Coordination test suite comprised prompts like "Who was tall? The happy stranger saw the angry men and the angry women were tall." More generally, the prompts followed the following template: "Who was ADJ1? The ADJ2 NN1 V the ADJ3 NN3 and the ADJ4 NN4 were ADJ1." We created 256 prompts by iterating through combinations of the words in Table 4. "None" adjectives were excluded from the text.

The QA model NP/VP suite comprised prompts like "Who had the telescope? The girl saw the boy with the telescope." The prompts followed the following template: "Who had the NN1? The ADJ1 NN2 ADV V the ADJ2 NN3 with the ADJ3 NN4." In this suite, the choice of V and NN4 was tightly coupled - one may see with a telescope but not see with a stick, for example. Table 5 details the combinations of words used to fill out the template, including V-NN4 pairs. Overall, we generated 256 prompts.

The QA model RC suite comprised prompts like "Who was desperate? The women and the men who were desperate bribed the politician." The prompts followed the following template: "Who was ADJ1? The ADJ2 NN1 and the ADJ3 NN2 who were ADJ1 V the NN3." We generated 192 example prompts by iterating over combinations of the words listed in Table 6, excluding sentences in which NN1 and NN2 or ADJ2 and ADJ3 would have been the same.

The Intervention suite for the QA model comprised prompts like "What was green? The human saw the keys by the cabinet which were green." More generally, prompts were created via the fol-

| NN1 | V1 | NN2 | V2 |
|---|---|---|---|
| (dog/child) | (scratched/bit) | (vet/girl/boy) | (ran/screamed/smiled) |
| author | wrote | book | grew |
| (doctor/professor) | lectured | student | listened |
| (girls/boys) | raced | (kids/children) | (watched/cheered) |
| (people/spectators) | watched | (show/movie) | (stopped/paused) |
| (lawyers/judges) | (studied/considered) | case | (languished/proceeded) |
| (people/viewers) | (notice/spot) | actor | (departs/stays) |
| (band/conventions) | left | (hotel/stalls) | closed |

Table 3: Words used for sentence generation in the Mask NP/Z test suite.

| Category | Words |
|---|---|
| ADJ1 | tall, short |
| ADJ2 | happy, None |
| ADJ3 | angry, None |
| ADJ4 | angry, None |
| NN1 | stranger, child |
| NN2 | men, women |
| NN3 | women, men |
| V | saw, believed |

Table 4: Words used for sentence generation in the QA Coordination test suite.

| Category | Words |
|---|---|
| V - NN4 | (saw, telescope), (poked, stick) |
| ADJ1 | tall, None |
| ADJ2 | short, None |
| ADJ3 | special, None |
| NN1 | man, woman |
| NN2 | boy, girl |

Table 5: Words used for sentence generation in the QA NP/VP test suite.

lowing template: "What was ADJ1? The NN1 V the NN2 by the NN3 which was/were ADJ1." By changing the plurality of NN2 or NN3 and replacing "was" with "were," the correct answer should change. Overall, we generated 288 sentences by iterating over all combinations of the words listed in Table 7, such that exactly one of NN1 and NN2 was plural at a time.

## C Hyperparameter Selection

In the intervention experiments in Section 4.3, we performed interventions at layer 4, based on results of a validation study shown below. We reported the results for probes with different dropout rates and for varying counterfactual losses, but we had to choose the layer of the QA model at which to perform interventions.

| Category | Words |
|---|---|
| ADJ1 | corrupt, desperate |
| ADJ2 | tall, smart, rich |
| ADJ3 | tall, smart, rich |
| NN1 | men, women |
| NN2 | men, women |
| NN3 | judge, politician |

Table 6: Words used for sentence generation in the QA RC test suite.

| Category | Words |
|---|---|
| ADJ1 | green, large, dirty |
| NN1 | human, stranger, child |
| NN2 | key, keys, gadget, gadgets |
| NN3 | cabinet, cabinets, vase, vases |

Table 7: Words used for sentence generation in the QA intervention experiments.

Therefore, we created a validation suite based on the Intervention template, using new nouns, verbs, and adjectives. For dropout rates from 0.0 to 0.3, ranging over counterfactual losses, and layers from 1 to 7, we computed the QA model's F1 and Exact Match scores on the validation suite. These results are included in Table 8, and strongly suggested that performance, for all probes, was most increased via interventions at layer 4.

## D Varying Dropout Rates

In the main paper, we reported included only some of the results for distance- and depth-based probe interventions. Here, we first show, in more detail, how increasing the dropout rate grows the causal effect with the QA attachment suite and distance probes of varying $\alpha$. Next, we include the mean causal effect plots for Mask and QA models using both types of probes on the 5 total suites.

| $\alpha$/Loss | Layer | 0.05 | 0.1 | 0.2 | 0.3 |
|---|---|---|---|---|---|
| Dist. 0.0 | 1 | 71.9/59.4 | 72.7/60.9 | 73.4/60.9 | 73.4/60.9 |
| | 2 | 69.5/56.3 | 71.9/60.9 | 71.9/60.9 | 71.9/59.4 |
| | 3 | 71.1/60.9 | 71.1/59.4 | 71.9/59.4 | 71.9/59.4 |
| | **4** | **71.9/62.5** | **72.6/60.4** | **71.9/59.4** | **73.4/60.9** |
| | 5 | 68.8/57.8 | 68.8/56.3 | 72.7/60.9 | 73.4/62.5 |
| | 6 | 68.8/57.8 | 69.5/59.4 | 71.9/60.9 | 72.7/62.5 |
| | 7 | 70.3/60.9 | 70.3/60.9 | 72.6/62.5 | 72.6/62.5 |
| Dist. 0.1 | 1 | 69.5/56.3 | 71.1/59.4 | 71.9/59.4 | 71.9/59.4 |
| | 2 | 68.8/60.9 | 70.3/60.9 | 69.3/59.4 | 71.1/59.4 |
| | 3 | 67.2/56.4 | 69.5/60.9 | 72.7/60.9 | 73.4/62.5 |
| | **4** | **75.8/64.1** | **72.7/60.9** | **72.7/60.9** | **72.7/60.9** |
| | 5 | 68.8/56.3 | 70.3/59.4 | 71.9/57.8 | 71.1/56.3 |
| | 6 | 75.0/59.4 | 72.7/60.9 | 73.4/62.5 | 73.4/62.5 |
| | 7 | 72.7/60.9 | 72.7/62.5 | 72.7/62.5 | 72.7/60.9 |
| Dist. 0.2 | 1 | 69.5/54.7 | 70.3/56.3 | 72.7/59.4 | 73.4/60.9 |
| | 2 | 73.4/60.9 | 74.2/59.4 | 74.2/62.5 | 74.2/62.5 |
| | 3 | 70.3/59.4 | 69.5/56.3 | 71.1/57.8 | 71.9/57.8 |
| | **4** | **74.2/65.6** | **75.0/65.6** | **75.8/65.6** | **75.0/64.1** |
| | 5 | 71.1/62.5 | 71.9/64.1 | 71.1/62.5 | 71.9/60.9 |
| | 6 | 73.4/62.5 | 71.8/59.4 | 74.2/62.5 | 74.2/62.5 |
| | 7 | 71.9/59.4 | 73.4/62.5 | 72.7/62.5 | 72.7/60.9 |
| Dist. 0.3 | 1 | 67.2/54.7 | 70.3/59.4 | 73.4/62.5 | 72.7/60.9 |
| | 2 | 68.8/60.9 | 71.1/60.9 | 72.7/62.5 | 71.9/60.9 |
| | 3 | 61.7/53.1 | 64.8/56.3 | 71.9/64.1 | 72.3/65.6 |
| | **4** | **67.2/59.4** | **71.9/64.1** | **75.0/65.6** | **75.8/65.6** |
| | 5 | 62.5/56.3 | 68.8/59.4 | 70.3/62.5 | 70.3/62.5 |
| | 6 | 71.1/62.5 | 711/64.1 | 70.3/60.9 | 71.1/60.9 |
| | 7 | 75.0/64.1 | 72.7/62.5 | 71.9/62.5 | 73.4/62.5 |

Table 8: Validation Coord. suite results (F1/Exact Match) using distance probes. For each probe type, we iterated over intervention layer and counterfactual loss value. The small validation suite was useful for rapid identification of good hyperparameter settings. All probes had the best performance at layer 4 (in bold).

First, we plotted an example of how increasing the dropout rate grew the causal effect in the QA attachment quite in Figure 8. We found that positive dropout values consistently outperformed probes with no dropout. Furthermore, for $\alpha$ ranging from 0.1 to 0.4, increasing the dropout rate seemed to increase the effect size. Considering only interventions at layer 2, for example, vanilla probes shifted model predictions by at most 2% for different parses; for probes with dropout 0.5, probabilities shifted by roughly 20%.

Finally, we included results for all dropout rates and counterfactual losses in Figures 9 and 10.

## E Probe Performance Metrics

In the main paper, we demonstrated the benefits of using dropout probes for creating counterfactual embeddings. One could hypothesize that the dropout enables better counterfactuals because the probes are prevented from overfitting to the training data. We found that that was not the case.

In Figure 11, we plotted probe performance metrics for the distance- and depth-based probes. For the distance probe, we reported the spearman correlation coefficient between predicted and actual pairwise distances between words in a sentence's parse tree. For the depth probe, we reported the accuracy of the probe in predicting the word at the root of the syntax tree. Both metrics were used in prior probing literature (Hewitt and Manning, 2019).

We found that, while using non-linear probes boosted probe performance compared to linear probes, adding dropout actually worsened probe

QA Model Causal Effect on Attachment Suites Using Dropout Distance Probes

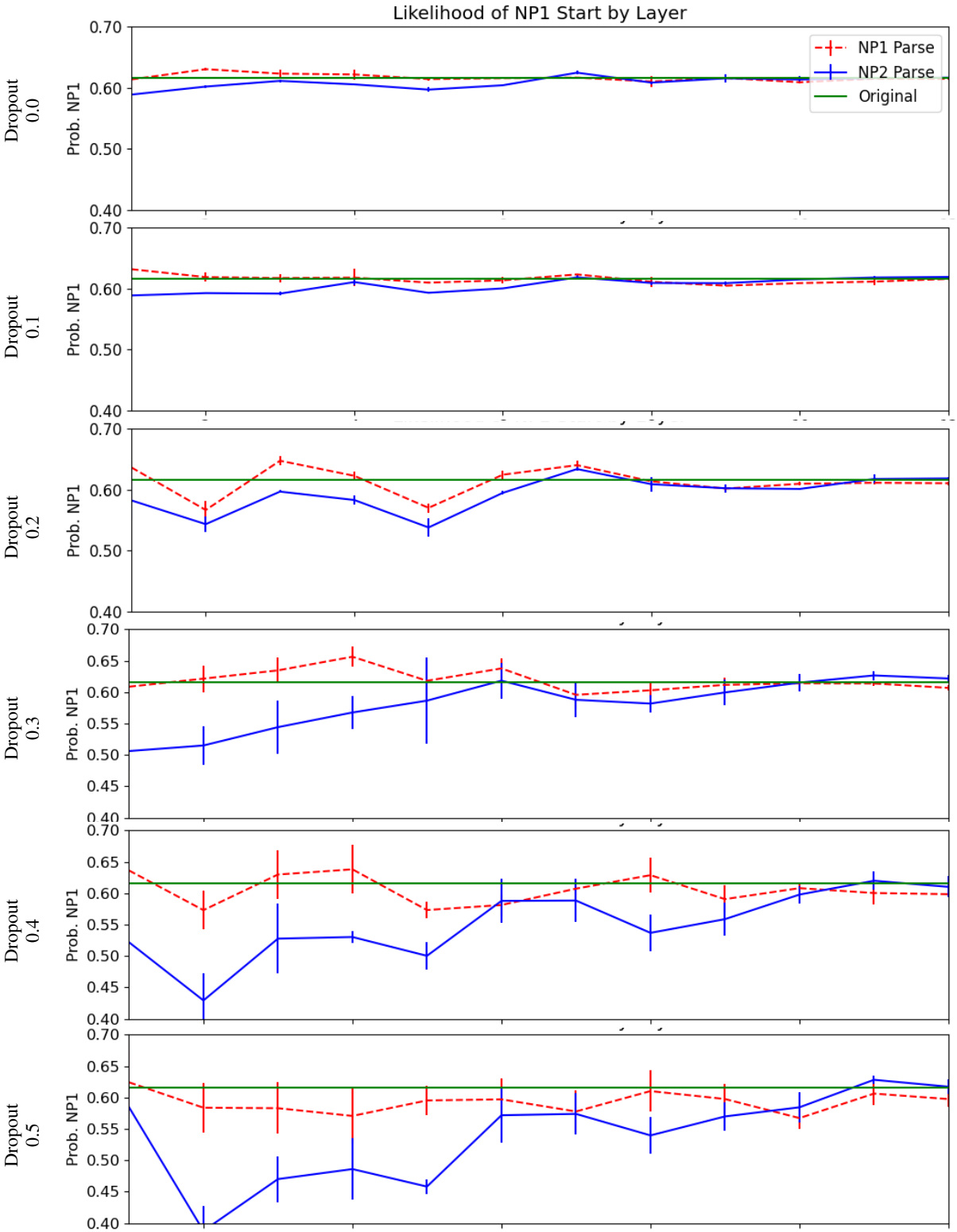



Figure 8: Dropout distance probes with dropout rates from 0.0 to 0.5 showed how, to a point, increasing the dropout rate increased the effect size for QA models on the Coord. suite.

performance. This suggests that the benefits from dropout in counterfactual generation arose from a phenomenon other than higher-performing probes.

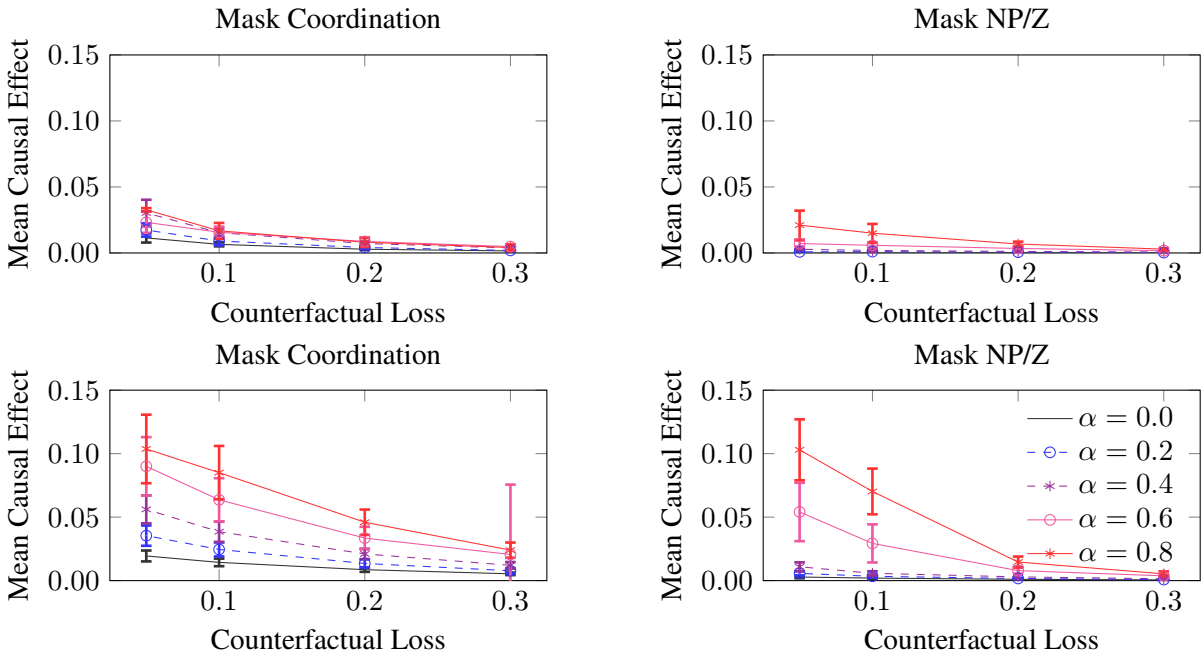



Figure 9: Mask mean causal effects using depth- (top) or distance-based (bottom) probes. Depth probes revealed smaller effects than distance-based probes, but a similar pattern of benefiting from lower counterfactual loss and higher dropout.

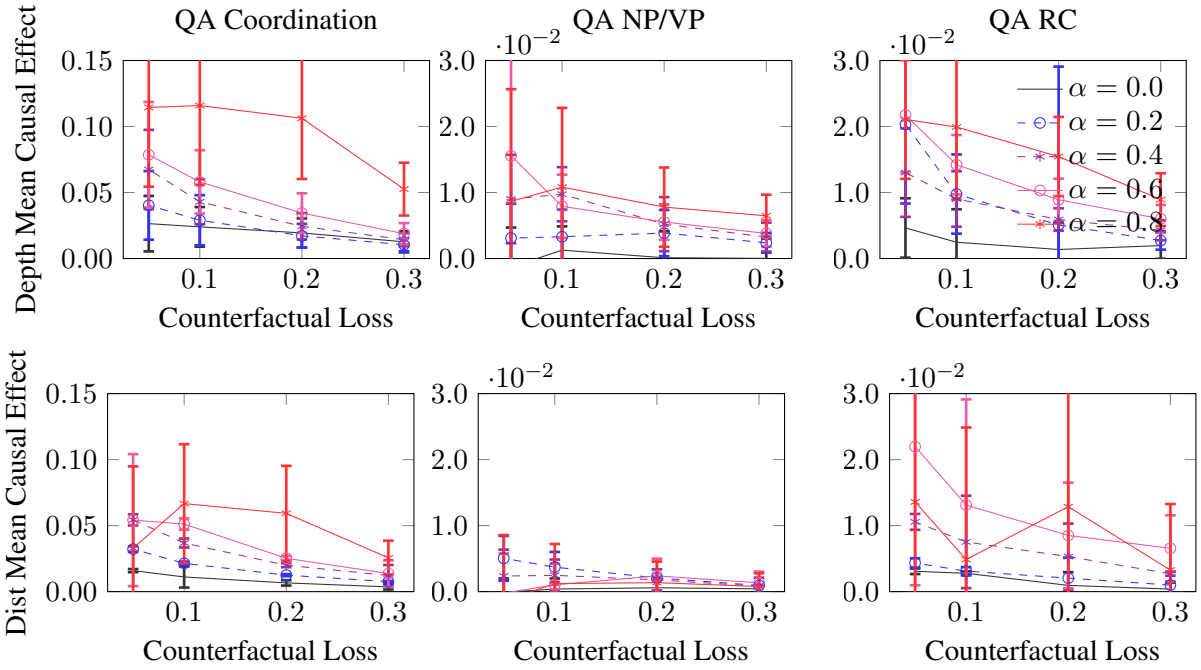


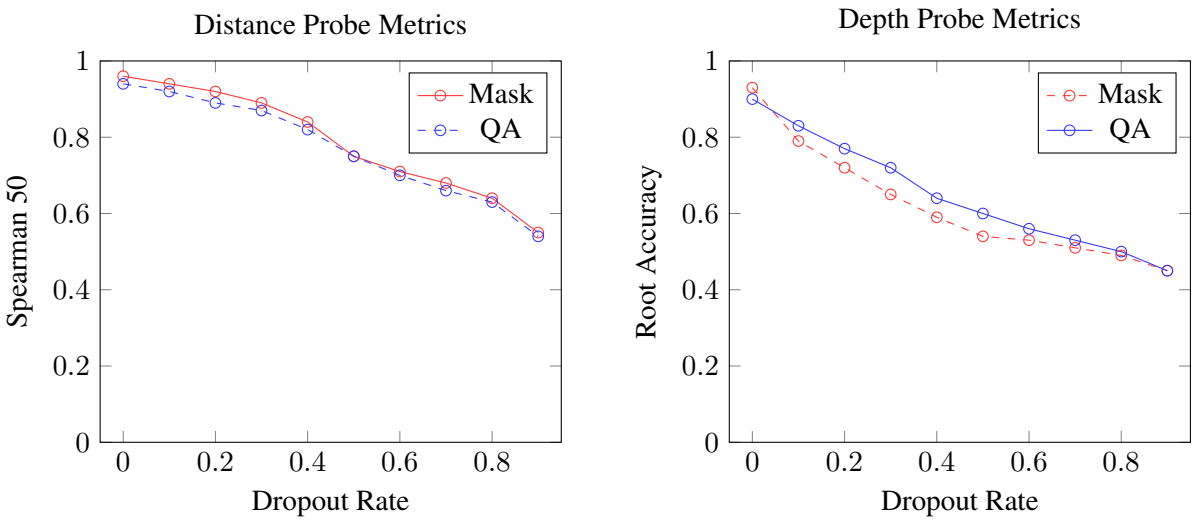

Figure 10: QA mean causal effects using depth-based (top) or distance-based (bottom) probes.

## F Scientific Artifacts

In this work, we built upon pre-existing scientific artifacts, including datasets and publicly-avaible code. Here, we briefly list their licenses and intended use cases. We used all artifacts for purely academic purposes.

The Penn TreeBank is licensed under the "LDC User Agreement for Non-Members" (Marcus et al., 1993). The dataset is commonly used in many academic settings (e.g., Hewitt and Manning (2019); Tucker et al. (2021)).

The Stanford Question Answering Dataset 2.0 is under a creative commons license and is commonly used in academic settings (Rajpurkar et al., 2016).




(a) Distance probe metrics

(b) Depth probe metrics

Figure 11: Metrics for the distance (left) and depth (right) probes showed that introducing dropout worsened probe performance as measured on the probe prediction tasks. Means over 5 trials plotted. All standard deviations less than 0.01.

The MNLI dataset is under an OANC license, "which allows all content to be freely used, modified, and shared under permissive terms" (Williams et al., 2018). The HANS dataset is under an MIT license (McCoy et al., 2019). Both datasets are commonly used in academic settings (McCoy et al., 2019).

The code we used to train the NLI and NLI-HANS models is under an Apache License 2.0 (Gao et al., 2021) and was developed for academic use.